# DAQ Data Structures

31-Oct-2012, Boo!
Eric Church w.
Cat James

# Data Record

- ❑Should be uniform for all datatypes that a user will pick up Offline.

- ❑Hence, downstream of Assembler.

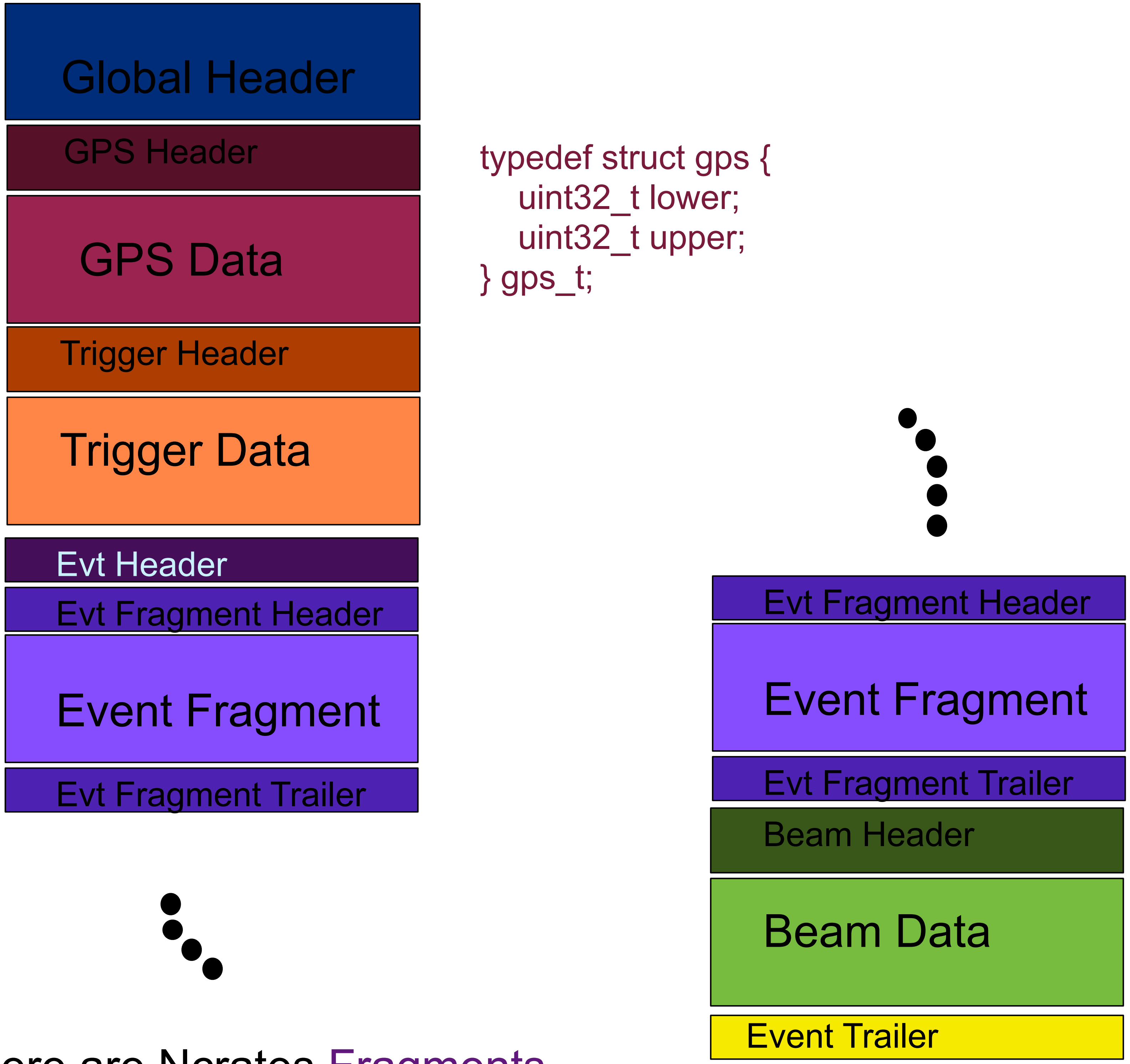
- ❑EventTypes: Calibration, Triggered, and (built!) SN

- ❑These 3 types will have been seen by Assembler.

# C++ Structs: Headers, data.

```
typedef struct {
  static const uint8_t DAQ_version_number = VERSION;
  uint8_t record_type;   /* From event_types.h */
  uint8_t record_origin; /* DATA or MC */
  uint32_t run_number;
  uint32_t event_number;
  uint32_t event_number_crate; /* Crate's sense of the evt #. */

    uint8_t spare8_2;
  uint8_t spare8_3;
 ...
  uint32_t number_of_bytes_in_the_record;
  uint32_t spare32_1;
} global_header_t;
```



```
typedef struct gps {
   uint32_t lower;
   uint32_t upper;
} gps_t;
```

Trigger Header,
GPS Header,
Evt Header,
Beam Header
contain simply
uint32_t sizeData

There are Ncrates Fragments.

# What types might we be forgetting?

- Laser, Strobe ... and Granite Block and Veto runs are proposed to be just variants of Triggered Events.
- Partial detector runs-- PMT only, or only a few TPC racks -- are read-outs, I propose, that should also go through assembler.
- That leaves un-built SN evts. Purple from all crates. And 10th crate contributes non-zero sized Trigger and GPS data.